

- b) die noch zu ladenden Konfigurationsdaten bestehender früherer Anforderungen aus einem Zwischenspeicher (FILMO) in die konfigurierbaren Elemente geladen werden,
- c) die bestehende Anforderung auf eine eindeutige Kennung (ID) umgesetzt wird
- 1.3 die ID auf die Adresse der zu ladenden Konfigurationsdaten im Speicher der Einheit umgesetzt wird, sofern die Konfigurationsdaten im Speicher der Einheit existieren,
- 1.4 die Konfigurationsdaten mit der entsprechenden ID bei einer übergeordneten Einheit angefordert und geladen werden, sofern die Konfigurationsdaten nicht im Speicher der Einheit existieren,
- 1.5 die Konfigurationsdaten in die konfigurierbaren Elemente geladen werden, sofern die konfigurierbaren Elemente die Daten annehmen können,
- 1.6 die Konfigurationsdaten der konfigurierbaren Elemente, die die Daten nicht annehmen können in einen Zwischenspeicher geladen werden,
- 1.7 nachdem die Konfigurationsdaten vollständig abgearbeitet sind, wieder neu auftretende Anforderungen akzeptiert werden, wobei bis zum Auftreten einer erneuten Anforderung die noch zu ladenden Konfigurationsdaten bestehender früherer Anforderungen aus einem Zwischenspeicher (FILMO) in die konfigurierbaren Elemente geladen werden.
- 2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß Konfigurationsdaten, die nicht an die konfigurierbaren Elementen angenommen worden sind, an das Ende des Zwischenspeichers geschrieben werden.
- 3. Verfahren nach Anspruch 1 und 2, dadurch gekennzeichnet, daß Konfigurationsdaten zum Schreiben in die konfigurierbaren Elemente vom Anfang zum Ende des Zwischenspeichers gelesen werden, wobei der Zwischenspeicher immer komplett durchlaufen wird.
- 4. Verfahren nach Anspruch 1 bis 3, dadurch gekennzeichnet, daß Konfigurationsdaten, die am Anfang des Zwischenspeichers in die konfigurierbaren Elemente geschrieben werden, im Zwischenspeicher gelöscht werden, d. h. der Lesezeiger wird hinter die geschriebenen Daten gesetzt.
- 5. Verfahren nach Anspruch 1 bis 4, dadurch gekennzeichnet, daß Konfigurationsdaten, die aus der Mitte des Zwischenspeichers in die konfigurierbaren Elemente geschrieben werden,
  - a) entweder im Zwischenspeicher als gelöscht markiert werden und damit bei einem erneuten Lesedurchlauf übersprungen werden,
  - b) oder im Zwischenspeicher (FILMO) gelöscht werden, indem die noch existierenden Konfigurationsdaten ausgehend von der zu löschenden Speicherstelle bis zum Anfang oder zum Ende des Zwischenspeichers (FILMOs) so verschoben werden, daß die gelöschte Speicherstelle mit bestehenden Konfigurationsdaten belegt ist und die Zeiger auf den Beginn oder das Ende des Speichers entsprechend angepaßt werden.
- 6. Verfahren nach Anspruch 1 bis 5, dadurch gekennzeichnet, daß eine Zustandsmaschine (Garbage-Kollektor) den Speicher (CTR) der Einheit so verwaltet, daß keine Speicherlücken entstehen, indem Speicherlücken mit existierenden Konfigurationsdaten ausgehend vom Beginn der Speicherlücke bis zum Ende des Speichers (CTR) so verschoben werden, daß die gelöschte Speicherstelle mit bestehenden Konfigurationsdaten belegt ist und die Zeiger auf das Ende des Speichers und die Übersetzung der IDs auf verschobene Speicherstellen entsprechend angepaßt werden.
- 7. Verfahren nach Anspruch 1 bis 6, dadurch gekennzeichnet, daß bei der Einheit eingehende Anforderungen auf eine eindeutige Kennung (ID) einer Konfigurationsroutine übersetzt wird, und die ID auf eine Speicherstelle im Speicher (CTR) zeigt.
- 8. Verfahren nach Anspruch 1 bis 7, dadurch gekennzeichnet, daß Konfigurationsdaten das Laden weiterer Konfigurationsdaten bewirken (EXECUTE).
- 9. Verfahren nach Anspruch 1 bis 8, dadurch gekennzeichnet, daß Konfigurationsdaten mittels Anforderung an eine andere Einheit das Ausführen weiterer Konfigurationsdaten in der anderen Einheiten bewirken (TRIGGER), wobei die Anforderungen
  - a) als Broadcast an alle anderen Einheiten gesendet werden,
  - b) nur an die direkt übergeordnete Einheit gesendet werden,
  - c) nur an die direkt untergeordnet(n) Einheit(en) gesendet werden
  - d) an eine bestimmte, adressierte Einheit gesendet werden.
- 10. Verfahren nach Anspruch 1 bis 9, dadurch gekennzeichnet, daß die Einheiten hierarchisch in einer Baumstruktur angeordnet sind.
- 11. Verfahren nach Anspruch 1 bis 10, dadurch gekennzeichnet, daß die höchste Einheit in der Baumstruktur einen gemeinsamen Speicher mit einem übergeordneten Rechner teilt und der übergeordnete Rechner die Abläufe in dem Baustein über Kommunikation mit der übergeordneten Einheit steuert.
- 12. Verfahren nach Anspruch 1 bis 11, dadurch gekennzeichnet, daß Einheiten mittlerer und höchster Hierarchieebene neben gewöhnlichen Anforderungen (Trigger) auch auf Anforderungen von IDs reagieren, wobei die Übersetzung der ID in die Adresse der Speicherstelle der referenzierten Konfigurationsroutine erfolgt und die Übersetzung eines Triggers in eine ID entfällt.
- 13. Verfahren zum Cachen von Daten und Befehlen in aus mehreren Rechenwerken bestehenden Mikroprozessoren und in Bausteinen mit zwei- oder mehrdimensionaler Zellanordnung (z. B. FPGAs, DPGAs, DFPs gemäß DE 44 16 881 A1, o. dgl.), dadurch gekennzeichnet, daß
  - 13.1 mehrere Zellen und Rechenwerke (CEL) zu einer Mehrzahl von Gruppen zusammengefaßt werden, wobei jeder Teilgruppe ein Cache (CT) zugeordnet ist,
  - 13.2 die Caches der einzelnen Teilgruppen werden über eine Baumstruktur an einen übergeordneten Cache (ROOT-CT) geschaltet, der Zugriff auf einen Speicher (ECR) besitzt, in dem die Daten/Befehle abgelegt sind,
  - 13.3 jeder untergeordnete Cache oder Cache auf mittlerer Ebene des Baumes fordert die benötigten Daten/Befehle bei dem jeweils übergeordneten Cache an,
  - 13.4 die Daten werden über einen Write-Through-Mechanismus an die übergeordneten CTs und in den Speicher (ECR) geschrieben werden,

- 13.5 die Daten werden während des Zugriffes durch eine KR/TKR in allen CTs durch Eintragen der ID der betreffenden KR/TKR in das CTR/ECR als "ungültig" markiert,  
 13.6 beim Rückschreiben der Daten wird die ID in dem zusätzlichen Eintrag des CTR/ECR gelöscht, wodurch die Daten als gültig markiert sind.
14. Verfahren zum Arbitrieren von Daten dadurch gekennzeichnet, daß 5
- 14.1 die zu arbitrierenden Signale über eine Maske auf einen Round-Robin-Arbitrer geleitet werden, wobei der Grundzustand der Maske alle Signale durchschaltet,
- 14.2 der Arbiterausgang gibt das arbitrierte Signal an,
- 14.3 das arbitrierte Signal wird so dekodiert und auf die Maske zurückgeführt, daß das Signal und alle anderen höherpriorisierten Signale von der Maske gesperrt werden, 10
- 14.4 ist das letzte Signal arbitriert oder es existiert kein von der Maske freigeschaltetes Signal, das eine Arbitrierung anfordert, wird die Maske komplett freigeschaltet.

Hierzu 21 Seite(n) Zeichnungen

15

20

25

30

35

40

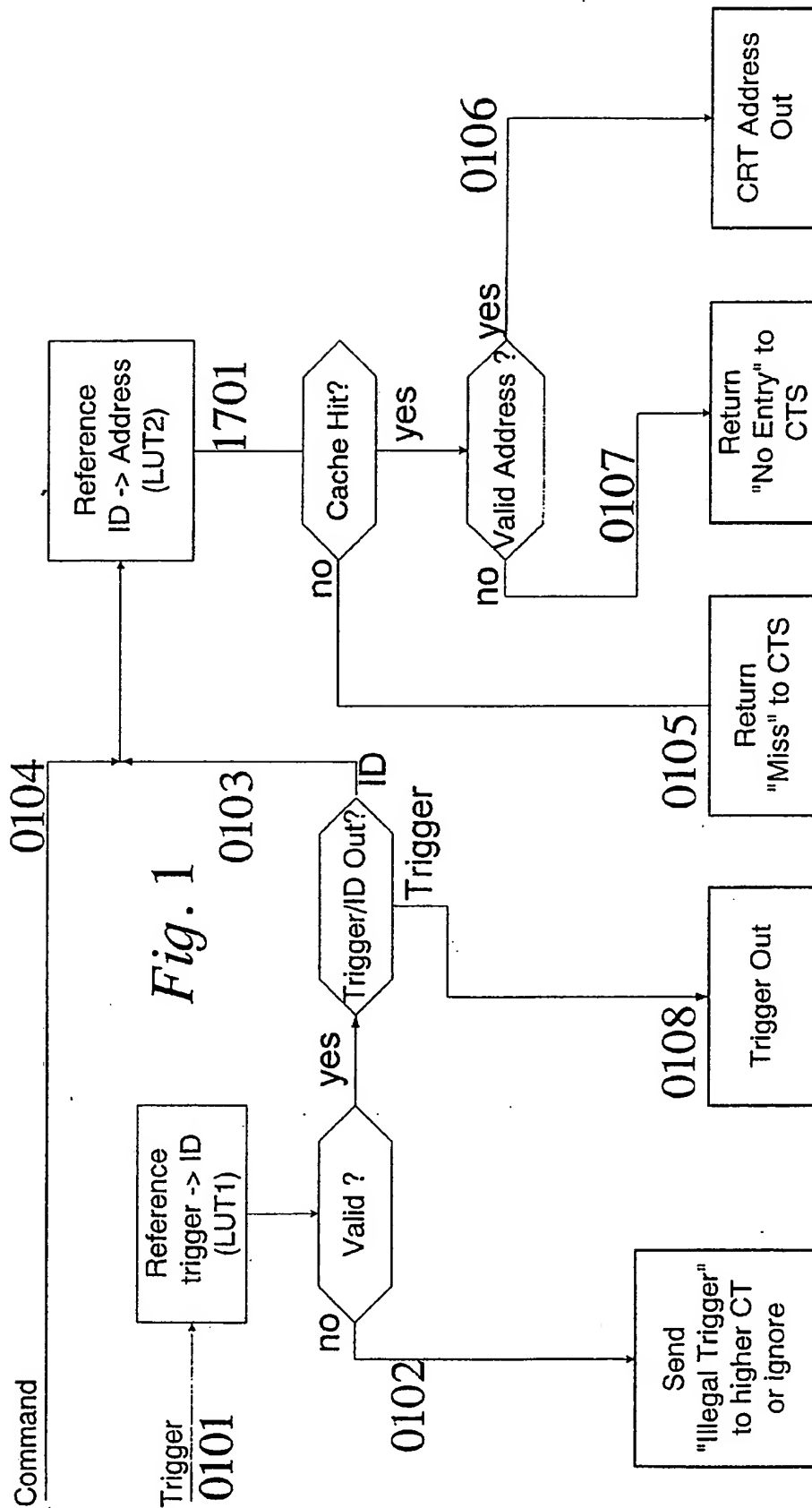
45

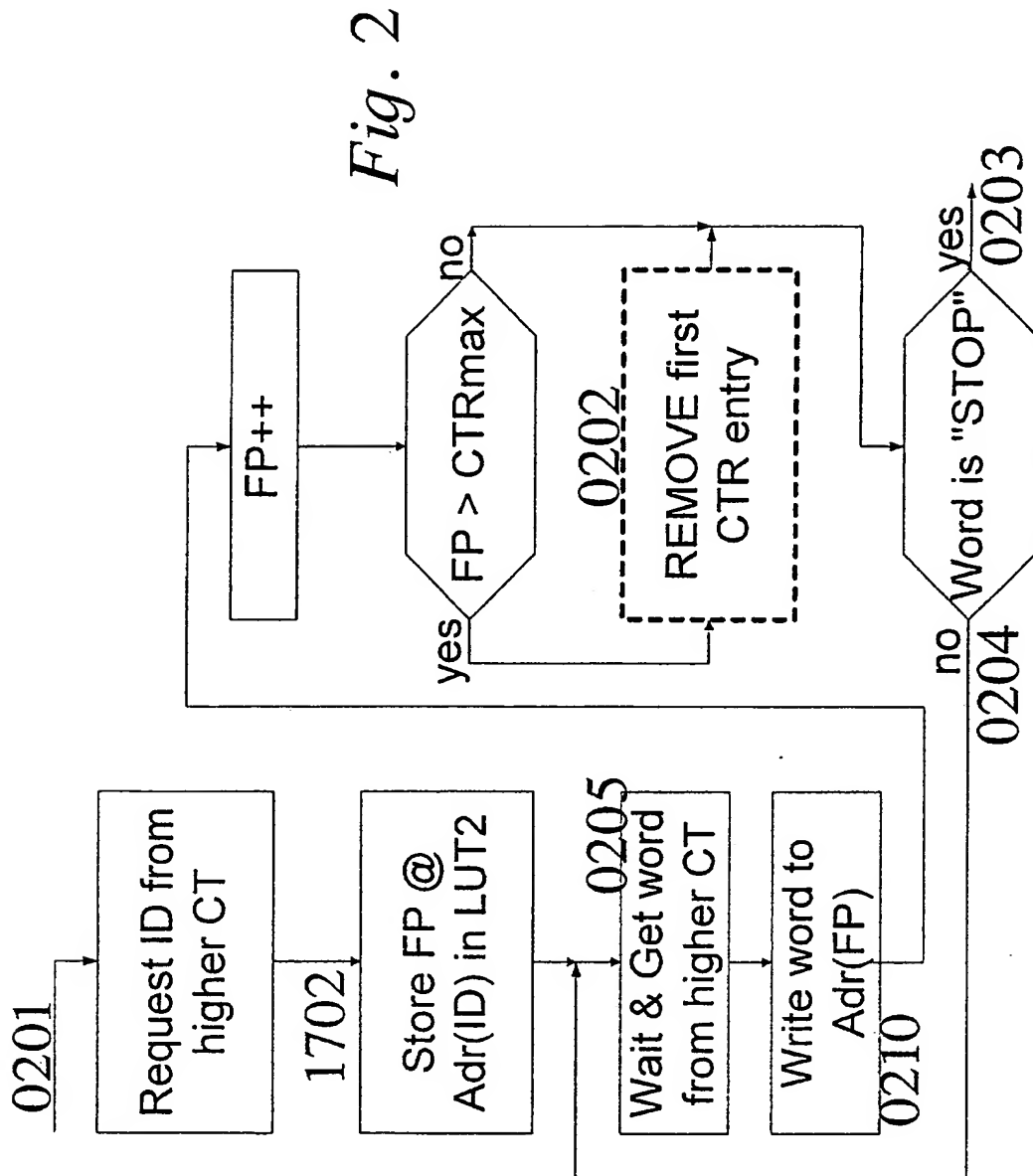
50

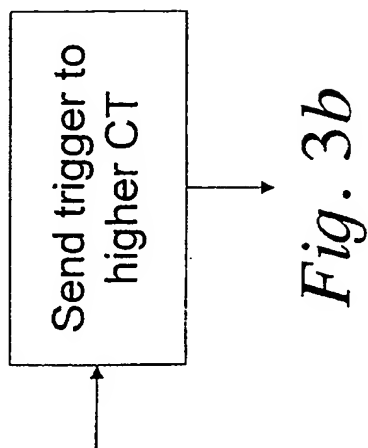
55

60

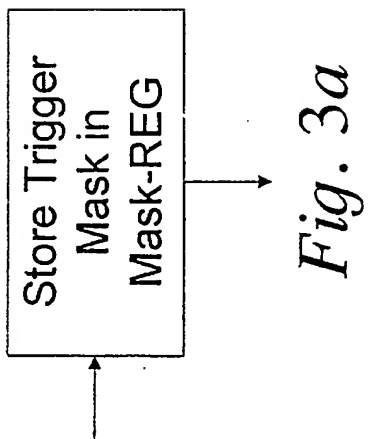
65



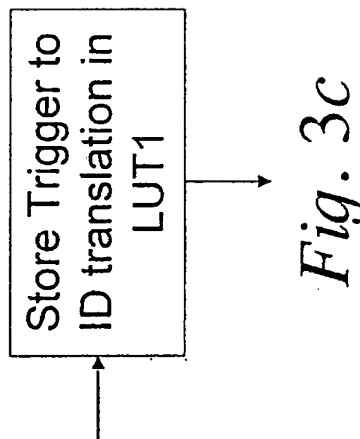




*Fig. 3b*



*Fig. 3a*



*Fig. 3c*

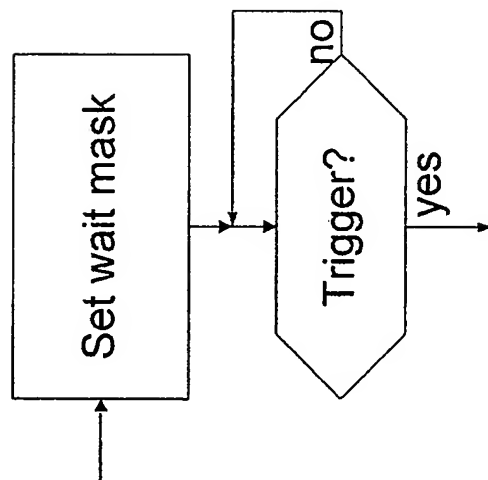
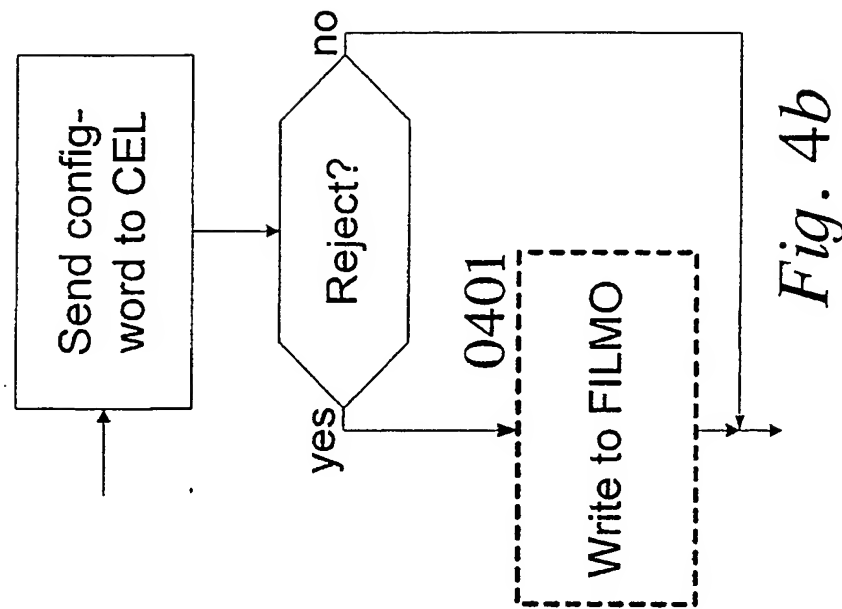


Fig. 5

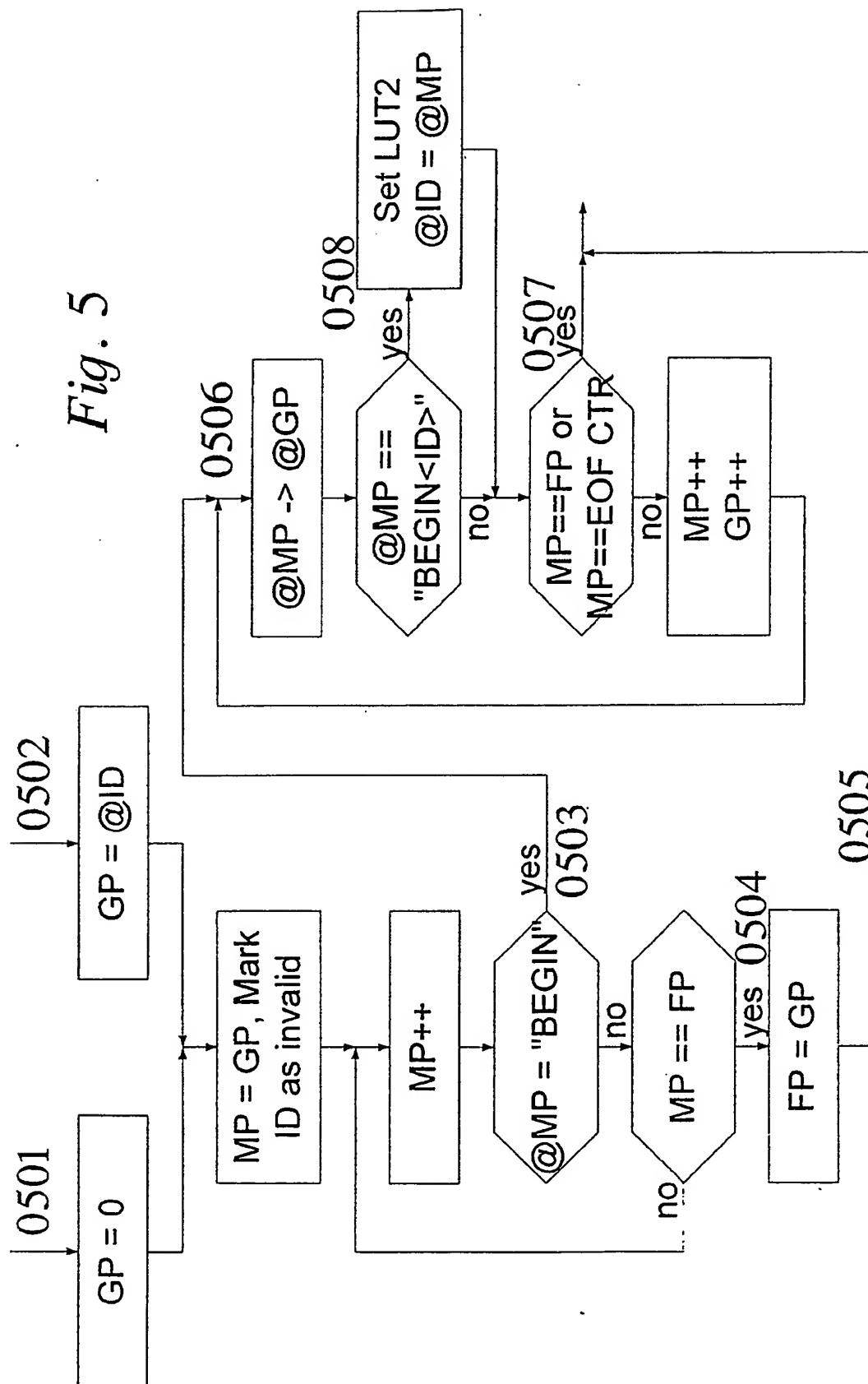
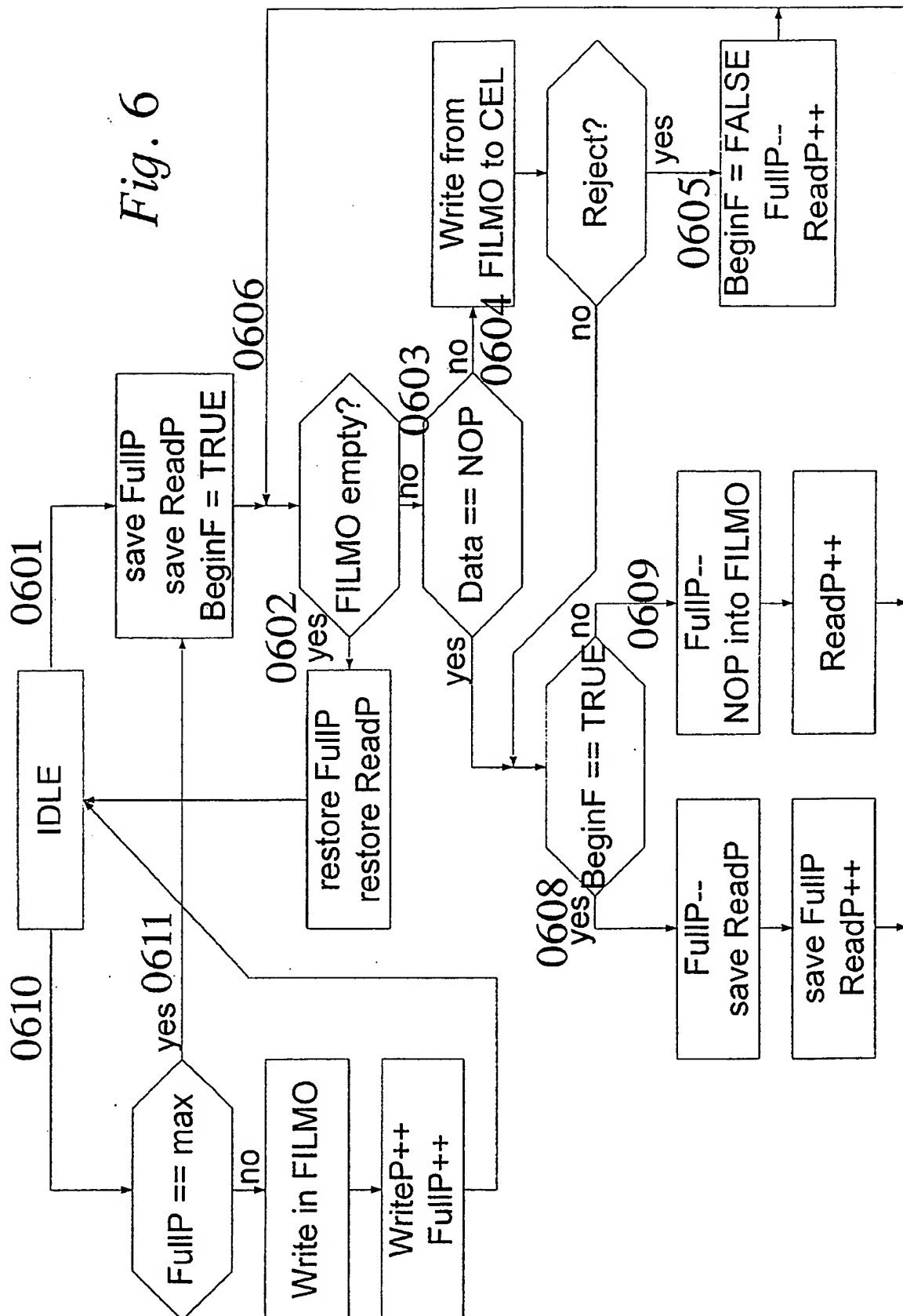
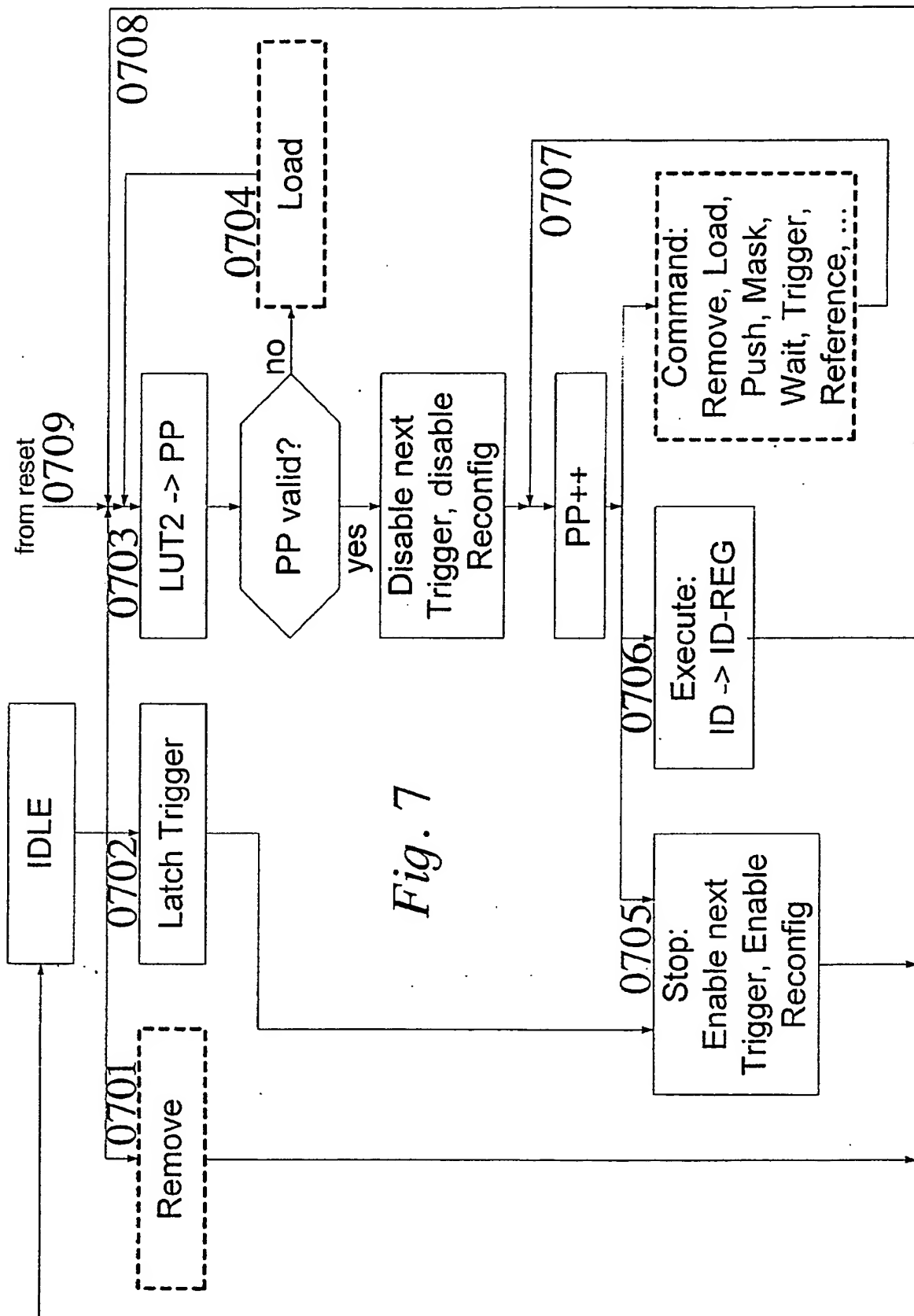


Fig. 6







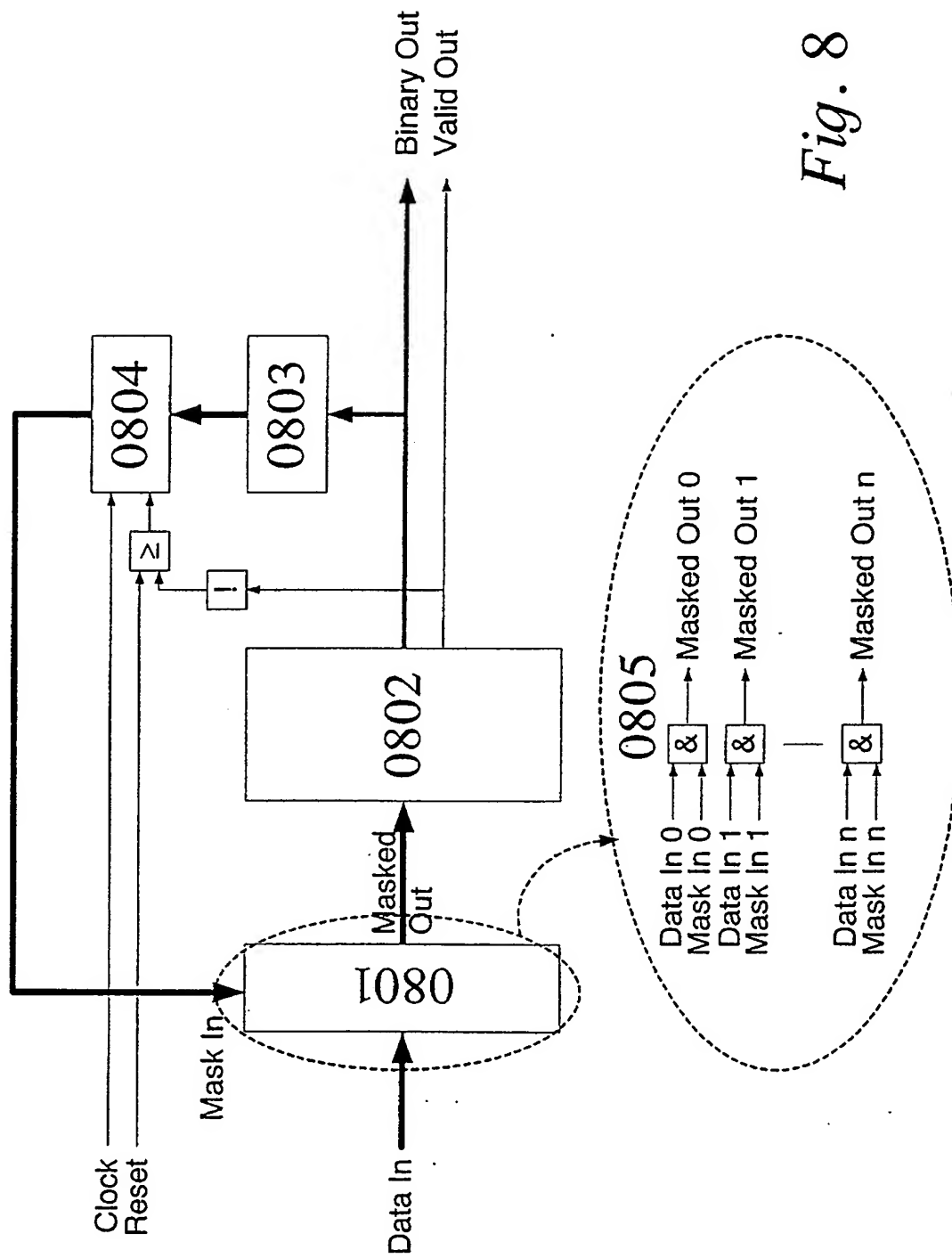
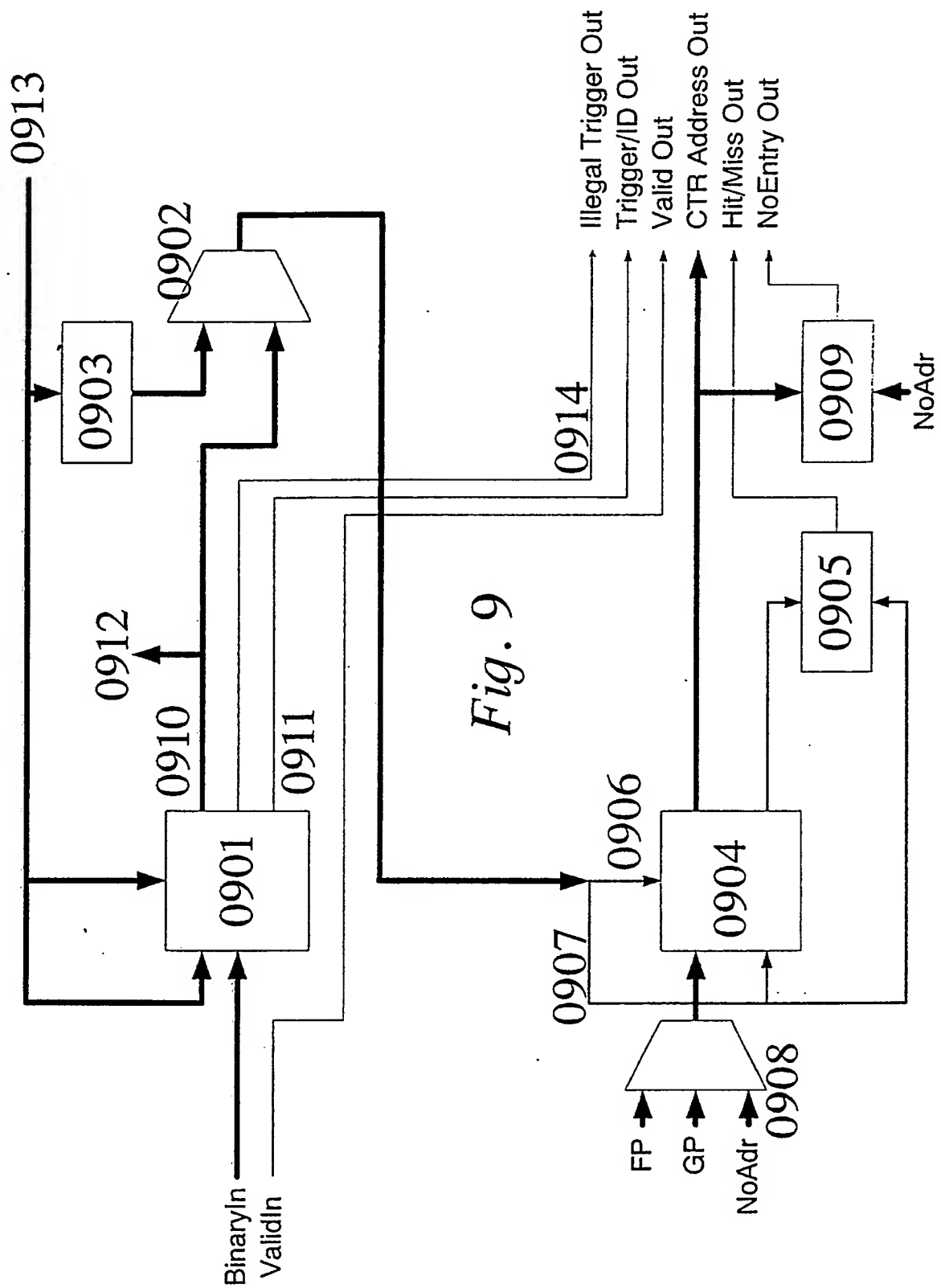


Fig. 8



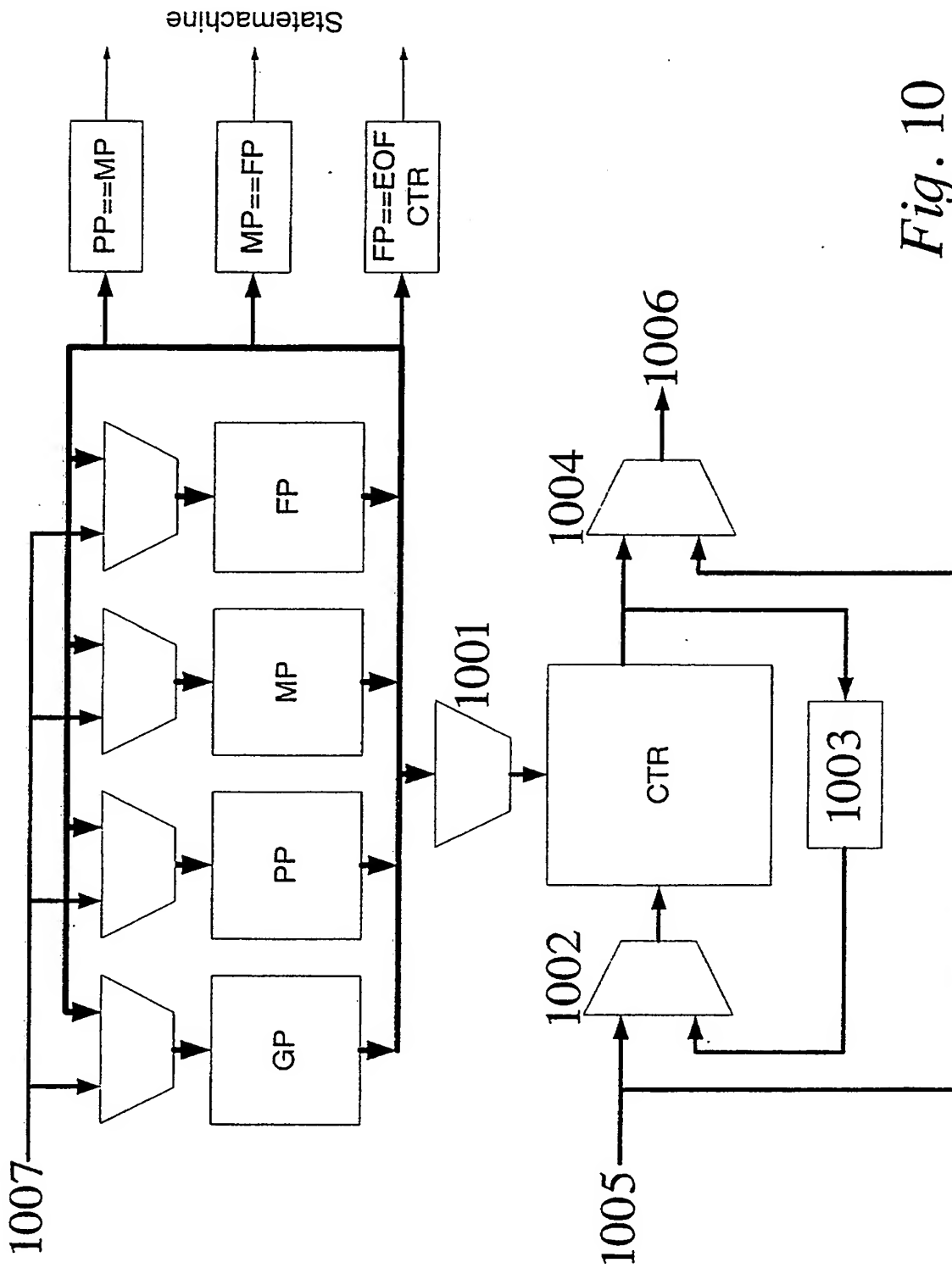
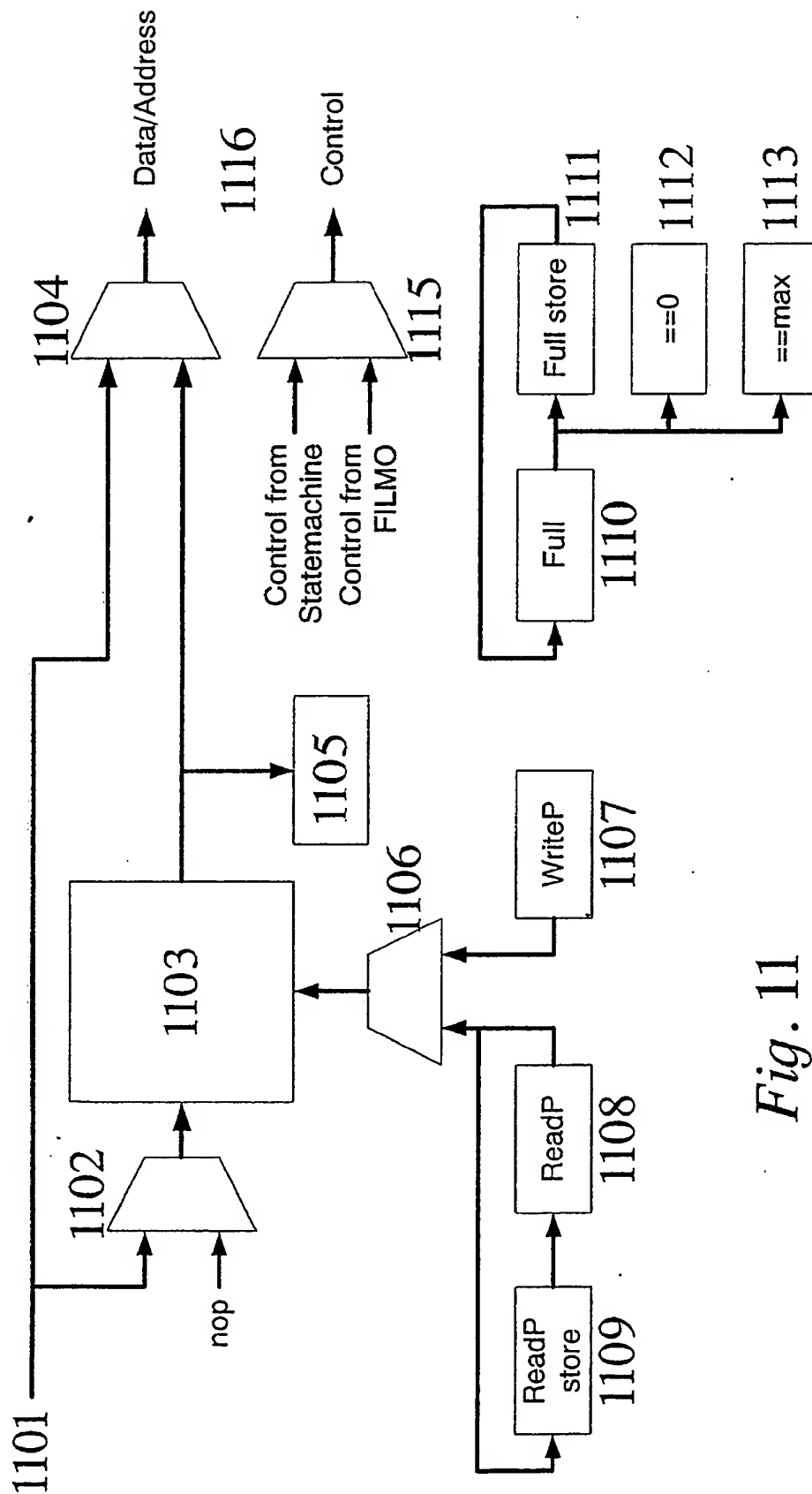


Fig. 10



*Fig. 11*

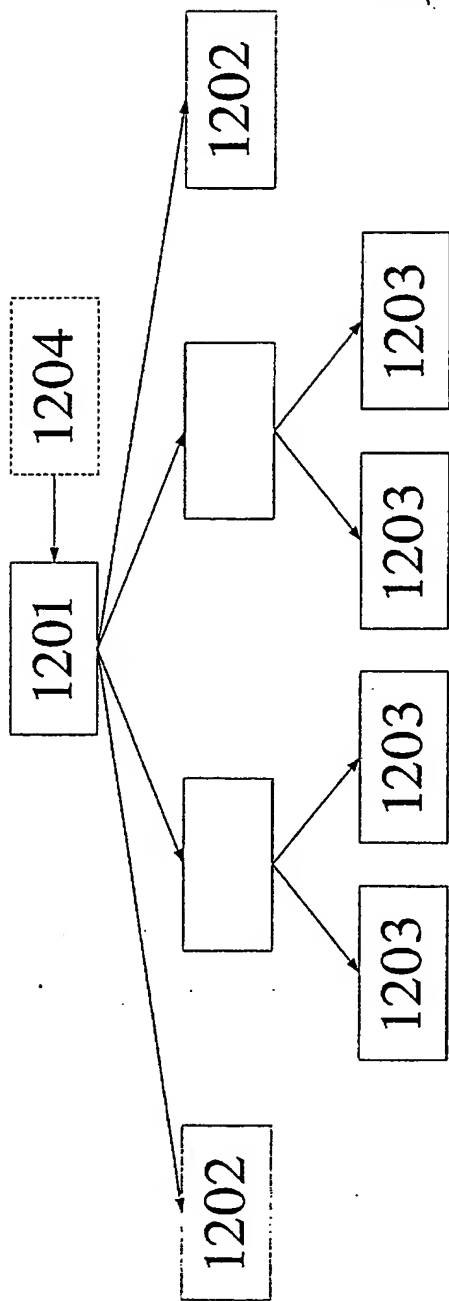


Fig. 12a

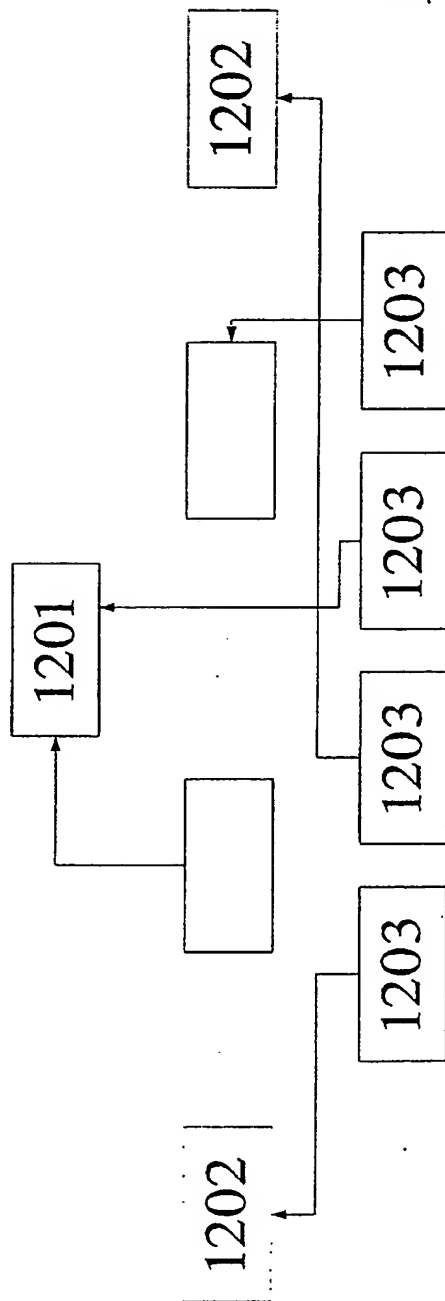
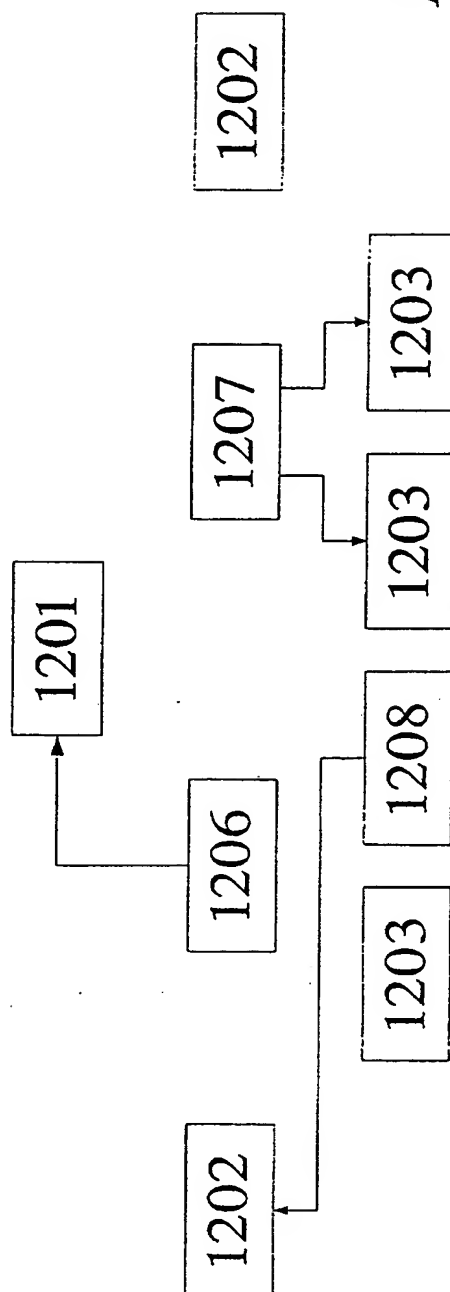
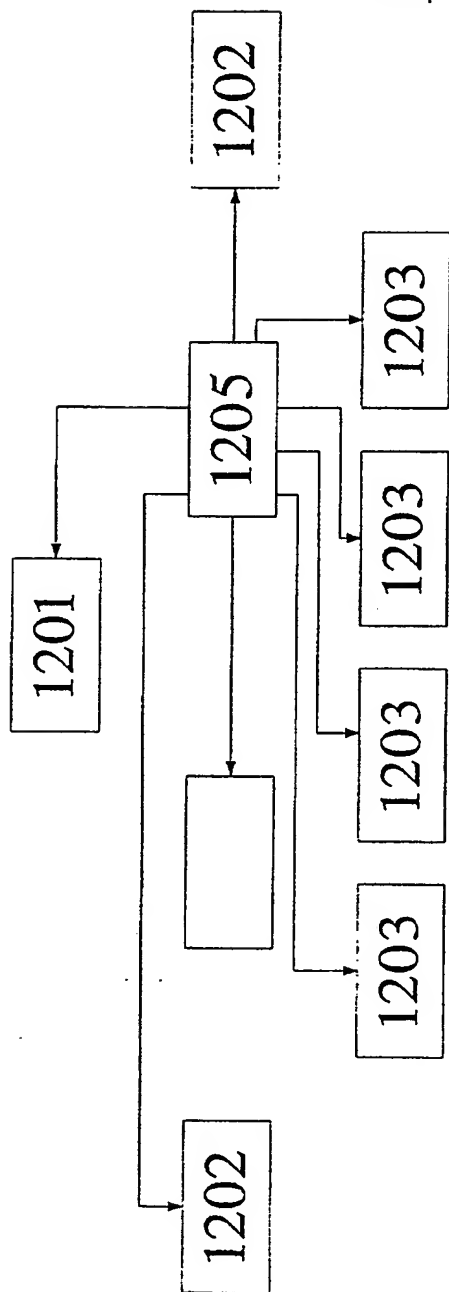
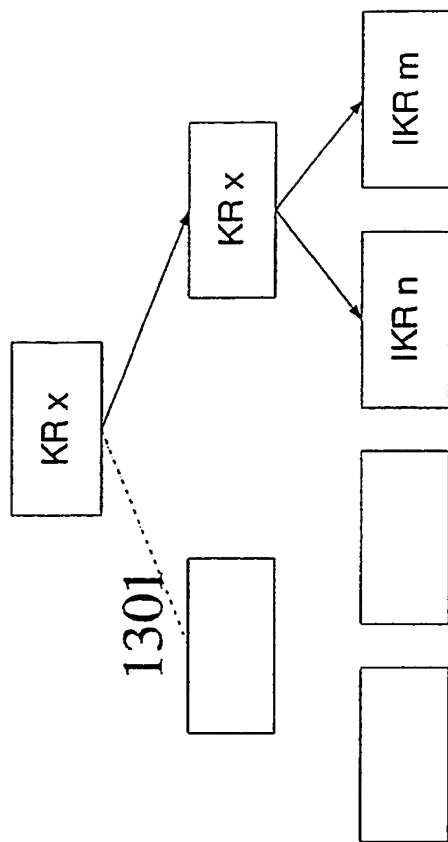


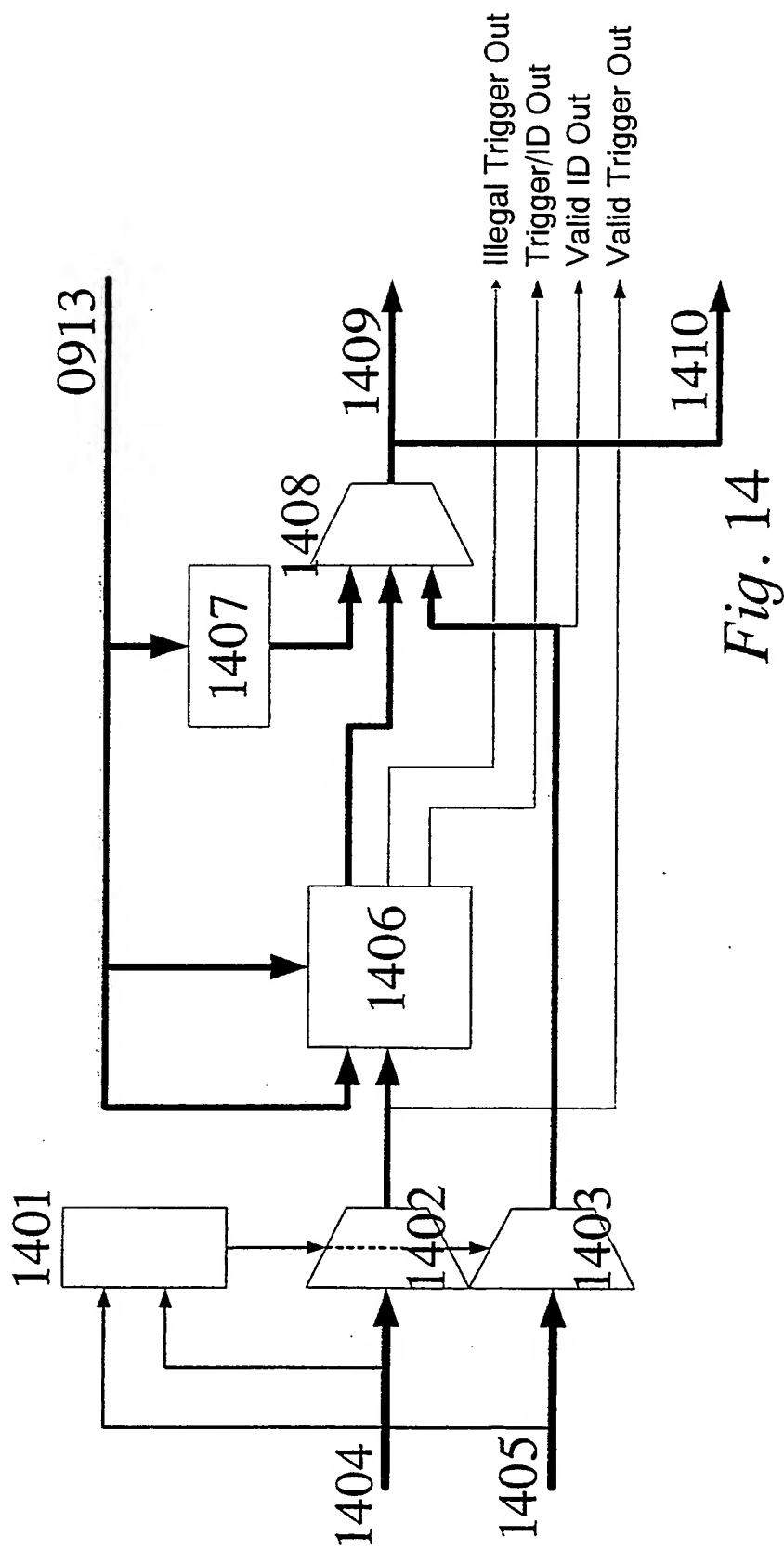
Fig. 12b

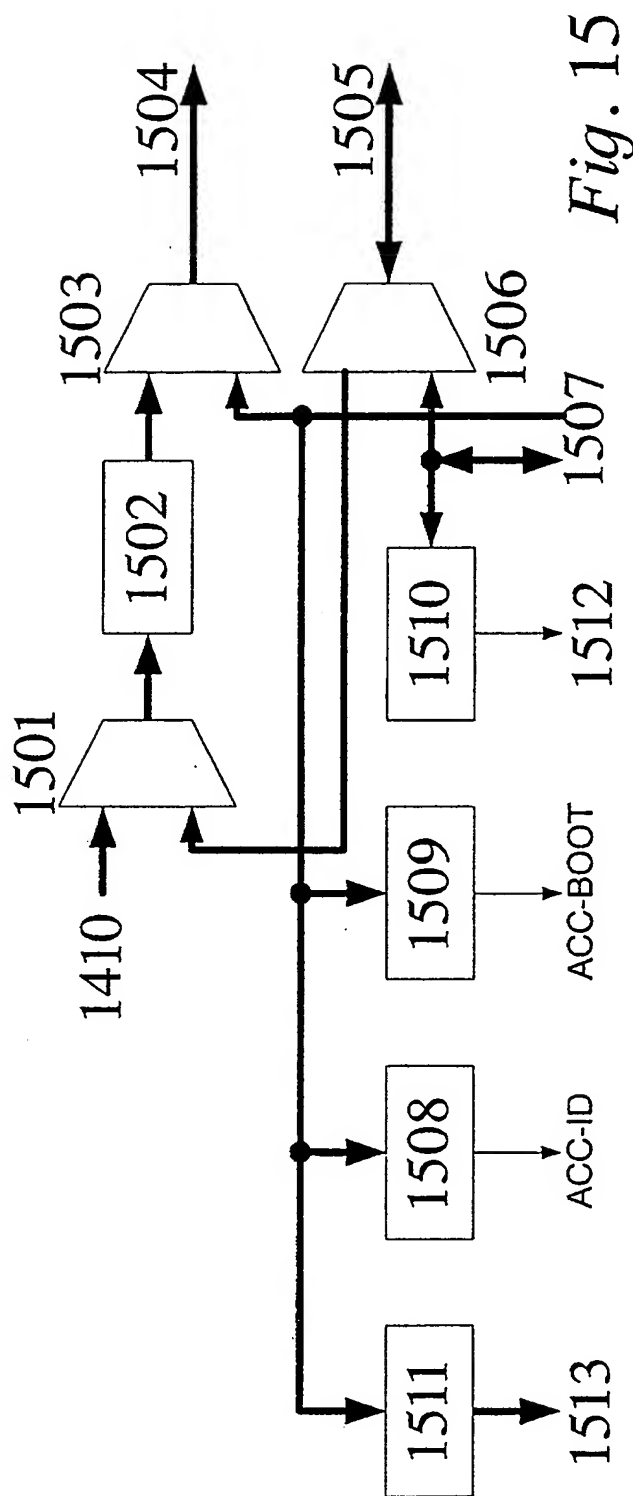




*Fig. 13*







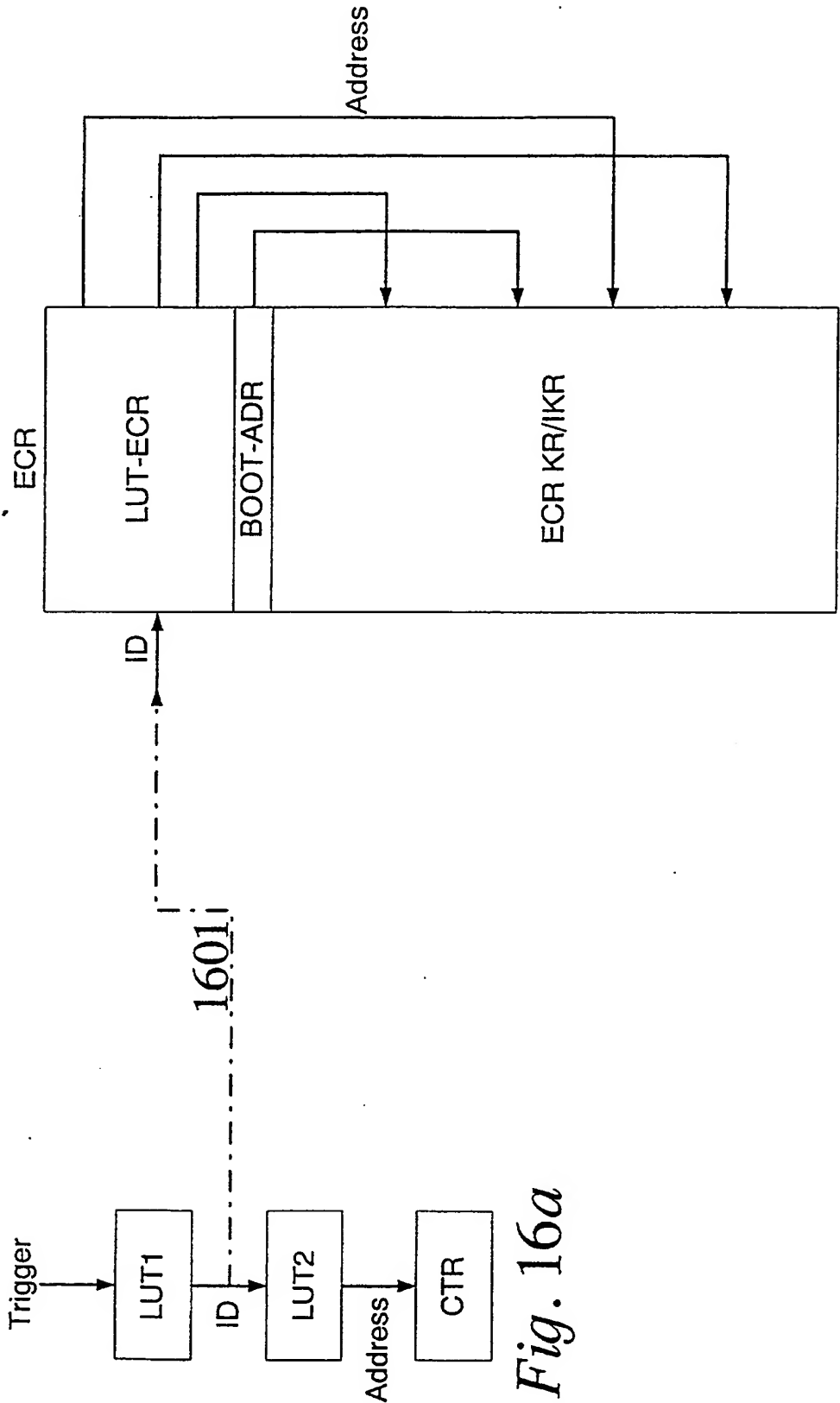


Fig. 16b

Fig. 16a

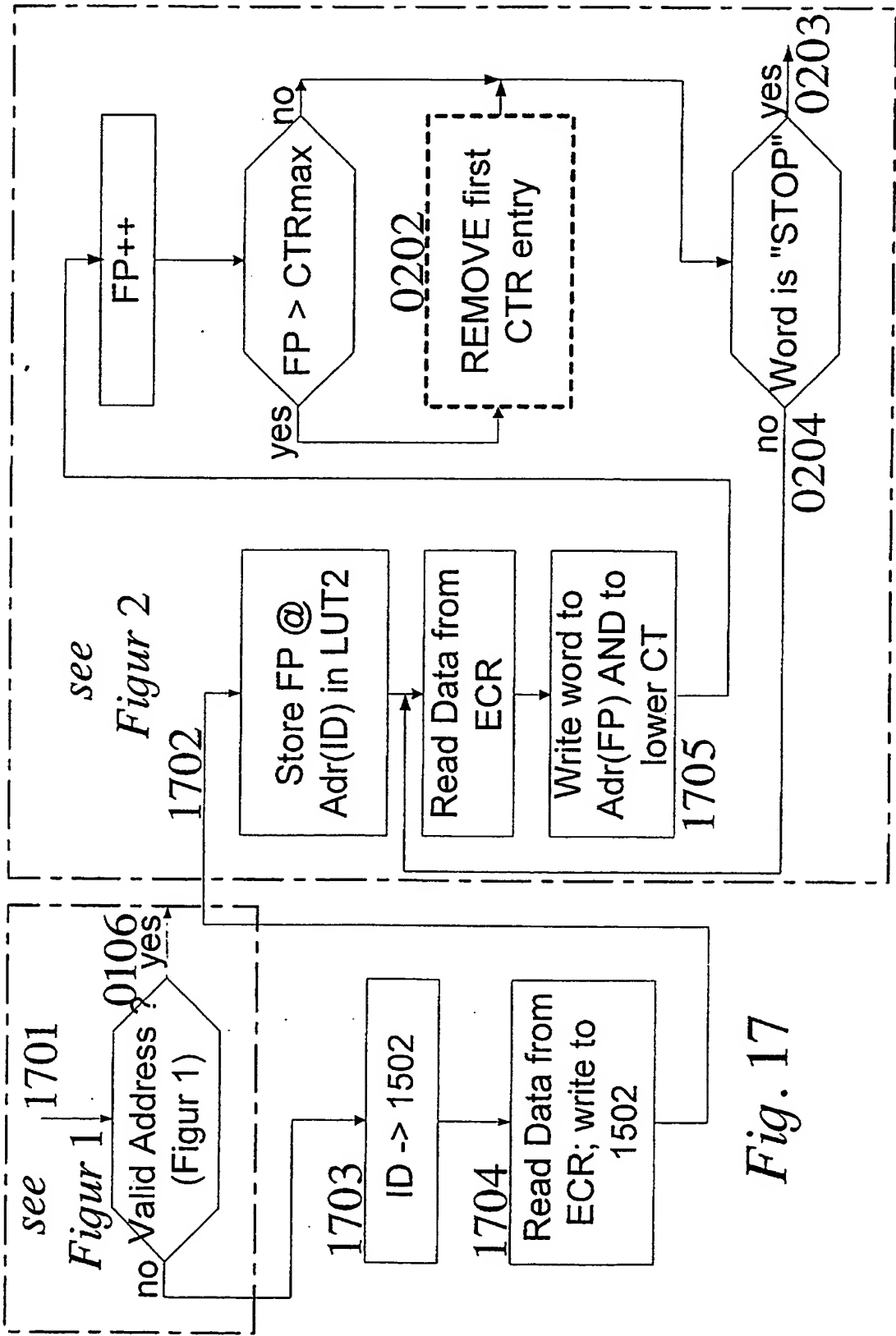
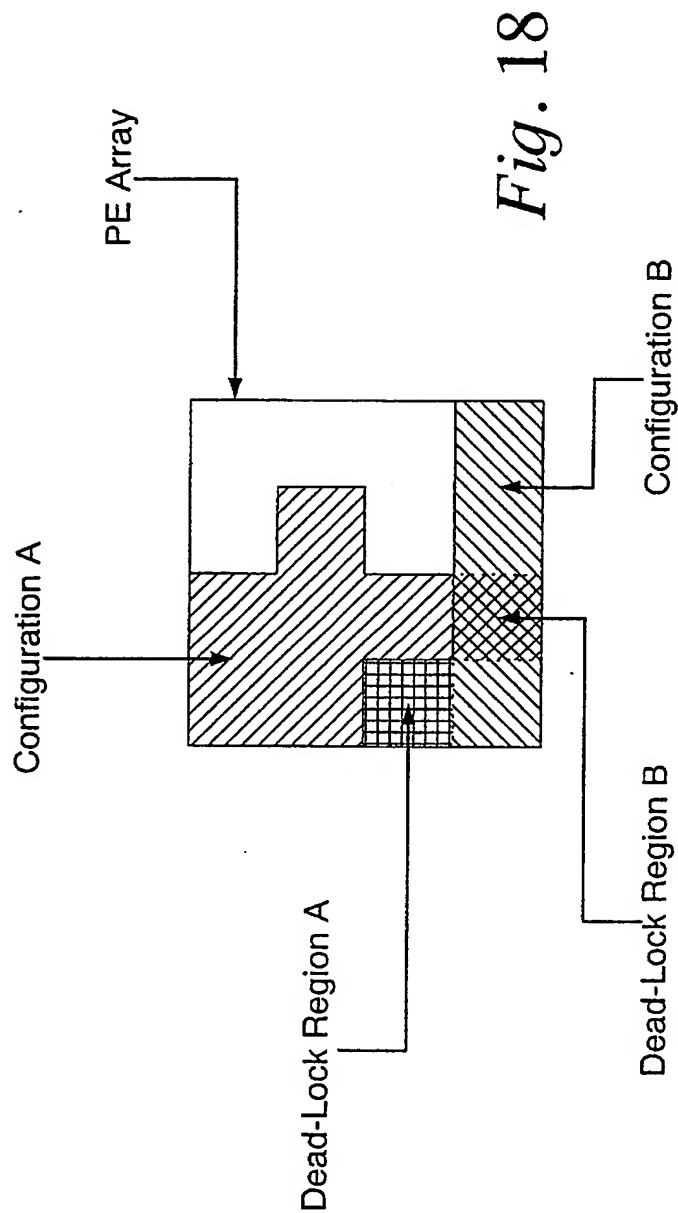


Fig. 17



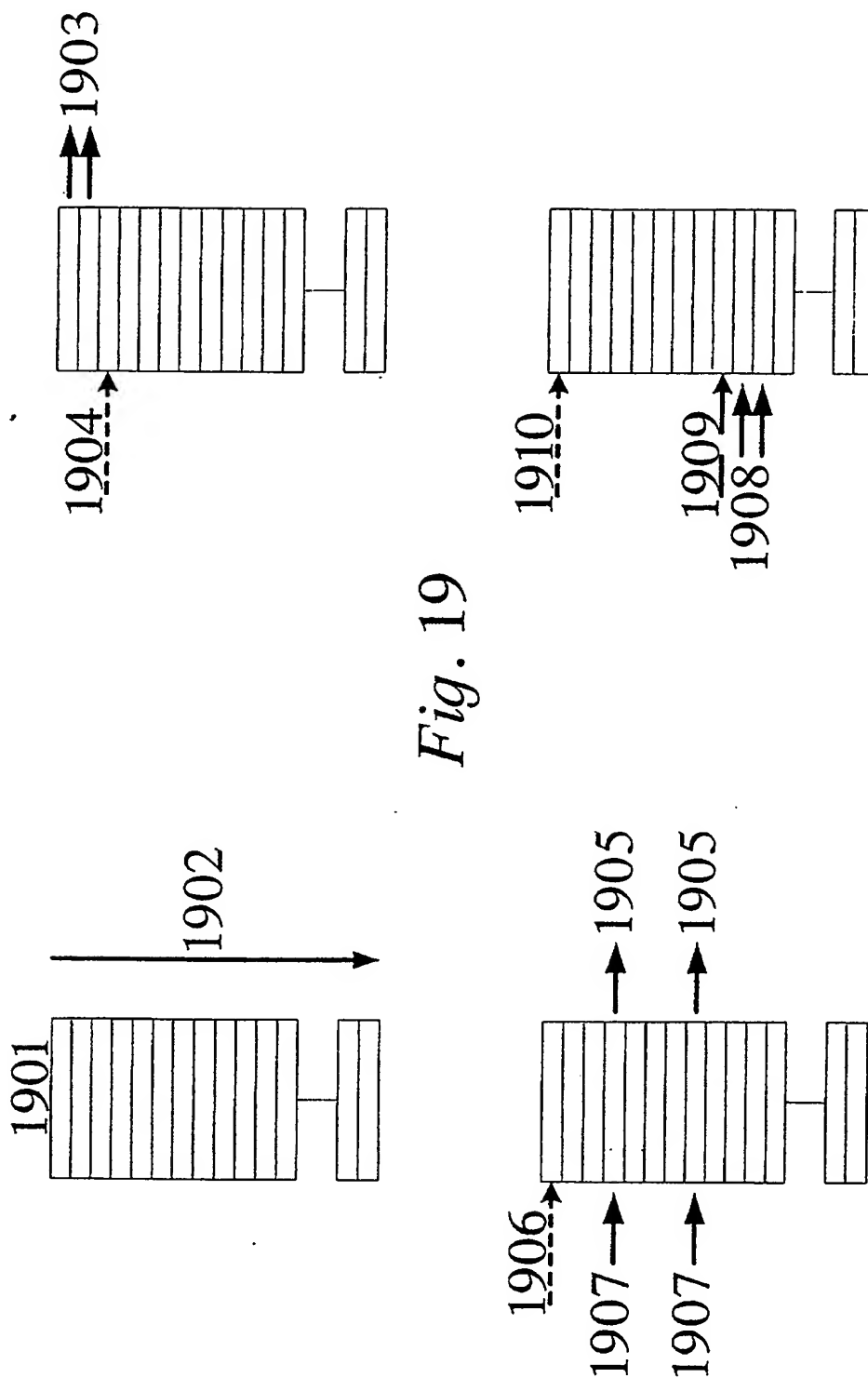


Fig. 19

